SERVER SECURITY

These are the steps required to harden a server before any production deployment (docker, ansible or command line)

1. Operating system patches up to date
2. Core dumps disabled
3. All Unnecessary services should be made unavailable
4. Disable system accounts
5. Strong PAM password quality
6. Password policy configured
7. SSH configuration hardened
8. No Unauthorized world-writable files
9. Mount options not hardened
10. SUID core dumps should be disabled
11. Dynamic network configuration
12. NFS and/or RPC services disabled
13. Networking hardening
14. IPv6 support disabled
15. Use TCPWrappers
16. Server should send logs to a remote LogHost
17. Strong log file permissions
18. Syslog and log rotation configured properly
19. Firewall

Operating system patches up to date

Using cron and aptitude

To begin, press Alt+F2 and create a new file:

```
gksudo gedit /etc/cron.weekly/apt-security-updates
```

Copy the following text into this new file, save, and exit:

```
echo "**************" >> /var/log/apt-security-updates
date >> /var/log/apt-security-updates
aptitude update >> /var/log/apt-security-updates
aptitude safe-upgrade -o Aptitude::Delete-Unused=false --assume-yes
--target-release `lsb_release -cs`-security >> /var/log/apt-security-updates
echo "Security updates (if any) installed"
```

Alternatively Install the `unattended-upgrades` package if it isn't already installed (`sudo apt-get install unattended-upgrades`).

To enable it, do:

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

Core dumps disabled

Core dump is a copy of process memory. Core dumps might reveal data that is not intended to be written to disk or disclosed to other users. Core dumps are not generally needed in production environments.

Core dump hard limit needs to be set to 0 with ulimit to properly restrict the creation of core dumps. To set limits permanently or for all processes, edit `/etc/security/limits.conf` Make sure the following config directive exists:

```
* hard core 0
```

and reboot.

All Unnecessary services should be made unavailable

Disable unnecessary services and limit the access to the services with firewalls when feasible. At minimum, make sure that firewall is configured and default action is to reject or drop traffic. Stateful rules should be used to allow return traffic. And if possible all administrative traffic should be allowed only from specified sources.

Disable system accounts

Login access from system accounts should be disabled. System accounts are not associated with a human user of the system, but exist to perform some administrative function. It is important to make sure that accounts that are not being used by regular users are locked to prevent them from logging in or running an interactive shell.

Accounts should be made less useful to an attacker by locking them and setting the shell to a shell not in /etc/shells. They can even be deleted if the machine does not use the daemon/service that account is responsible for, though it is safest to simply deactivate them. To deactivate them, lock the password and set the login shell to an invalid shell. /dev/null is a good choice because it is not a valid login shell, and should an attacker attempt to replace it with a copy of a valid shell the system will not operate properly

Strong PAM password quality

Configure pam_cracklib to require at least one uppercase character, lowercase character, digit, and other (special) character.

Enforce password lockout by configuring the number of attempts permitted before the account is locked to 6 and the time before the account is automatically unlocked to 900 seconds. Configure the system to use the SHA-512 algorithm. Do not allow users to reuse recent passwords, configure the number of remembered recent passwords to 12.

Example of applying recommended configuration in file /etc/pam.d/system-auth:

```
# require at least one uppercase character, lowercase character,

# digit, and other (special) character

password required pam_cracklib.so try_first_pass retry=3 minlen=14 \

dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1

# use the SHA-512 algorithm, remembered 12 recent passwords

password required pam_unix.so sha512 remember=12
```

attempts permitted before the account is locked to 6

and the time before the account is automatically unlocked to 900 s

auth required pam_tally2.so deny=6 onerr=fail unlock_time=900

account required pam_tally2.so

Password policy configured

Users should be forced to change their passwords, in order to decrease the utility of compromised passwords. However, the need to change passwords often should be balanced against the risk that users will reuse or write down passwords if forced to change them too often. Forcing password changes every 90-360 days, depending on the environment, is recommended.

Edit the file /etc/login.defs to specify password expiration settings for accounts. Add or correct the following lines:

PASS_MAX_DAYS 60

PASS_MIN_DAYS 7

PASS_MIN_LEN 8

PASS_WARN_AGE 7

SSH configuration hardened

Certain changes should be made to the OpenSSH daemon configuration file /etc/ssh/sshd config. The following recommendations can be applied to this file. See the sshd config(5) man page for more detailed information

Only SSH protocol version 2 connections allowed

Protocol 2

Disable .rhosts files

IgnoreRhosts yes

Disable host-based authentication

HostbasedAuthentication no

Disable root login via SSH

PermitRootLogin no

Disable empty passwords

PermitEmptyPasswords no

No Unauthorized world-writable files

Data in world-writable files can be modified by any user on the system. In almost all circumstances, files can be configured using a combination of user and group permissions to support whatever legitimate access is needed without the risk caused by world-writable files.

It is generally a good idea to remove global (other) write access to a file when it is discovered. However, check with documentation for specific applications before making changes. Also, monitor for recurring world-writable files, as these may be symptoms of a misconfigured application or user account.

Use following command to remove global write access from file:

chmod o-w file

Mount options not hardened

The nodev option prevents users from mounting unauthorized devices on any partition which is known not to contain any authorized devices.

Edit the file /etc/fstab file. The important columns for purposes of this section are column 2 (mount point), column 3 (filesystem type), and column 4 (mount options). For any line which satisfies all of the conditions:

the filesystem type is any of ext[234] and the mount point is not /

add the text ',nodev' to the list of mount options in column 4.

SUID core dumps should be disabled

See point 2

Dynamic network configuration

The Dynamic Host Configuration Protocol (DHCP) allows systems to request and obtain an IP address and many other parameters from a server. It is recommended that sites avoid DHCP as much as possible. Since DHCP authentication is not well-supported, DHCP clients are open to attacks from rogue DHCP servers. Such servers can give clients incorrect information (e.g. malicious DNS server addresses) which could lead to their compromise. If the machine must act as a DHCP client or server, configure it defensively.

For each interface IFACE on the system (e.g. eth0), edit /etc/sysconfig/network-scripts/ifcfg-IFACE and make the following changes:

1. Correct the BOOTPROTO line to read: BOOTPROTO=static

2. Add or correct the following lines, substituting the appropriate values based on your site's addressing scheme:

NETMASK=255.255.255.0

IPADDR=192.168.1.2

GATEWAY=192.168.1.1

NFS and/or RPC services disabled

NFS and RPC protocols are considered weak. They should be disabled if they are not absolutely necessary.

If there is no mission-critical reason for the machine to operate as either an NFS client or an NFS server, disable subsystems required by NFS. If the machine does not run any RPC-based services then disable the RPC portmapper service. Use chkconfig command to disable unnecessary services.

Networking hardening

Network parameters for hosts and routers improve Linux's ability to defend against certain types of IPv4 protocol attacks.

The accept_source_route, accept_redirects, and secure_redirects options are turned off to disable IPv4 protocol features which are considered to have few legitimate uses and to be easy to abuse.

The net.ipv4.conf.all.log_martians option is enabled to log several types of suspicious packets, such as spoofed packets, source-routed packets, and redirects.

The icmp_echo_ignore_broadcasts icmp_ignore_bogus_error_messages options protect against ICMP attacks.

The tcp_syncookies option uses a cryptographic feature called SYN cookies to allow machines to continue to accept legitimate connections when faced with a SYN flood attack.

The rp_filter option enables RFC-recommended source validation. It should not be used on machines which are routers for very complicated networks, but is helpful for end hosts and routers serving small networks.

IPv6 support disabled

As with any networking protocol, IPv6 should be disabled unless needed. Despite configuration that suggests support for IPv6 has been disabled, link-local IPv6 address auto configuration occurs even when only an IPv4 address is assigned. The only way to effectively prevent execution of the IPv6 networking stack is to prevent the kernel from loading the IPv6 kernel module.

To disable IPv6 support for all interfaces add following lines to /etc/sysctl.conf:

```
# IPv6

net.ipv6.conf.all.disable_ipv6 = 1
```

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1

net.ipv6.conf.eth0.disable_ipv6 = 1

Use TCPWrappers

By limiting access to the server, you reduce your exposure to threats from attackers on remote systems. For Internet-connected servers that provide service to the whole Internet, limiting access may not make sense. Intranet servers, limited-access servers, and workstations should limit access to only authorized networks. Many daemons (SSH for example) are compiled with TCP Wrapper support, so you can use /etc/hosts.allow and /etc/hosts.deny to limit SSH access to your systems. It is important to note that TCP wrappers looks at hosts.allow first, then hosts.deny, and controls access based on the first match. If you omit entries in hosts.allow and deny access to ALL in hosts.deny, you will block network access to all network clients.

The recommended setting is to deny anything not explicitly allowed. This is done by adding the following line in /etc/hosts.deny:

ALL: ALL

Then, explicitly list in /etc/hosts.allow all hosts/domains you want access to your machine.

Server should send logs to a remote LogHost

Remote logging is essential in detecting intrusion and monitoring. An intruder – once he/she has obtained root – can edit the system logs to remove all traces of the attack. If the logs are stored off the machine, those logs can be analyzed for anomalies and used for prosecuting the attacker.

Check in /etc/rsyslogd.conf file

Strong log file permissions

It is critical to protect system log files from being modified by unauthorized individuals. Also, certain logs contain sensitive data that should only be available to the system administrator. Configure file permissions according to recommended values

Syslog and log rotation configured properly

Successful local or network attacks on systems do not necessarily leave clear evidence of what happened. It is necessary to build a configuration in advance that collects this evidence, both in order to determine that something anomalous has occurred, and in order to respond appropriately.

To ensure that all important messages are captured consider to apply the configuration suggested by "Guide to the Secure Configuration of Red Hat Enterprise Linux 5":

Edit the file /etc/rsyslog.conf. Add or correct whichever of the following lines are appropriate for your environment:

auth,user.* /var/log/messages

kern.* /var/log/kern.log

daemon.* /var/log/daemon.log

syslog.* /var/log/syslog

lpr,news,uucp,local0,local1,local2,local3,local4,local5,local6.* /var/log/unused.log

Verify that each log file referenced in /etc/rsyslog.conf is also rotated - listed in the /etc/logrotate.d/syslog file.

Make sure that there is reasonable history of log data available - 2 months is recommended

Firewall

UFW, or Uncomplicated Firewall, is an interface to `iptables`. Assuming default ports we should do

```
$ sudo ufw allow http
$ sudo ufw allow https
$ sudo ufw allow ssh
```

Specify the port if using non standard like

```
$ sudo ufw allow 2222
```

Then enable these changes

```
sudo ufw enable
```